intel®

INTELLEC® SERIES IV OPERATING AND PROGRAMMING GUIDE

[illegible]

Order Number: 135840-002

INTELLEC® SERIES IV OPERATING AND PROGRAMMING GUIDE

Order Number: 135840-002

CAUTION

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause interference to radio communications. It has been tested and found to comply with the limits for a Class A Computing Device pursuant to Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

AEDIT	iDIS	iPDS	OpenNET
BITBUS	iLBX	iRMX	Plug-A-Bubble
BXP	$\mathrm{i}_{\blacksquare}$	iSBC	PROMPT
COMMputer	iMDDX	iSBX	Promware
CREDIT	iMMX	iSDM	QueX
Data Pipeline	Insite	iSXM	QUEST
Genius	Intel	KEPROM	Ripplemode
i	$\mathrm{int_{e}l}$	Library Manager	RMX/80
$\underset{\mathrm{I}}{\Delta}$	$\mathrm{int_{e}l}$BOS	MCS	RUPI
iATC	Intelevision	Megachassis	Seamless
I^{2}ICE	$\mathrm{int_{e}}$ligent Identifier	MICROMAINFRAME	SLD
ICE	$\mathrm{int_{e}}$ligent Programming	MULTIBUS	SOLO
iCS	Intellec	MULTICHANNEL	SYSTEM 2000
iDBP	Intellink	MULTIMODULE	UPI
	iOSP		

REV.	REVISION HISTORY	DATE
-001	Original Release	09/85
-002	Removes "Release 3.0" from Manual Title	01/86

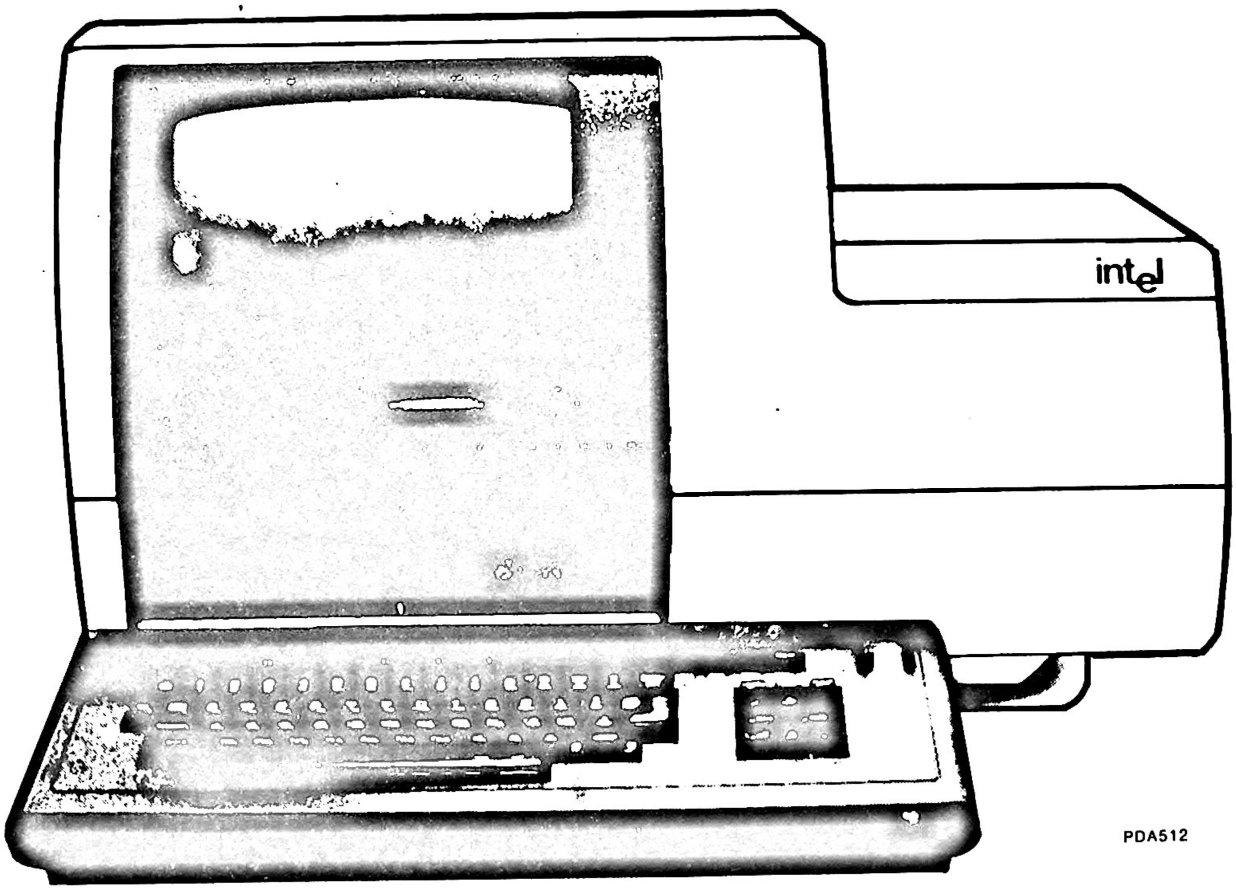
PREFACE

This manual provides operating and programming instructions for the Intellec Series IV Development System. The Series IV has two operating environments: one for 8086/8088-based software and one for 8080/8085-based software. Chapters 1-5 comprise the operations guide; 6 and 7 comprise the programmers reference manual.

This manual is designed to support new development system users as well as those who are already familiar with Intellec development systems. This manual assumes that you have read the *Intellec® Series IV Microcomputer Development System Overview* and are familiar with the terms listed in its glossary.

This manual has seven chapters and six appendices:

- Chapter 1, "Introduction," introduces you to the Series IV Development System.
- Chapter 2, "Human Interface," describes the terminal, editing capabilities, device names, console operations, menu selection, the command line interpreter, command files, and job control functions.



Series IV Development System (Front View)

- Chapter 3, "Operation and Management," describes the iNDX hierarchical file structure, directory files, file access and ownership, and file system considerations.
- Chapter 4, "Command Descriptions," describes the Series IV commands and gives examples for storing, identifying, and manipulating your files and jobs.
- Chapter 5, "Network Operation," describes the NDS-II and remote job control, and operating your Series IV in the NDS-II Network.
- Chapter 6, "Programming Introduction," describes operating system considerations and target environments, and lists the built-in service routines.
- Chapter 7, "The 8086/8088-Based Environment," defines the conceptual considerations and external procedures for the Series IV service routines in an 8086/8088-based environment.
- Appendix A, "CLI Command Syntax," lists the syntax of the commands detailed in Chapter 4.
- Appendix B, "Parameters and System Service Routines," is a condensed version of the routines detailed in Chapter 7.
- Appendix C, "Error Messages and Exception Codes," lists the various error messages generated by the operating system and the UDI interface.
- Appendix D, "Object Module Relocation and Linkage," defines the possibilities for combining object modules.
- Appendix E, "Boot Device-Configuration Switch Assignments," shows the Configuration switch settings necessary to boot the operating system from various physical devices.
- Appendix F, "ASCII Codes," shows ASCII codes, their meanings, and their decimal, octal, and hexadecimal values.

Required Software

The Series IV is an Intellec Microcomputer Development System that provides support for both development and execution of programs using either the 8086/8088 chip or the 8080/8085 chip. The Series IV contains both hardware and software enhancements not available with earlier versions of Intellec development systems.

The system contains:

iNDX	—	the 8086/8088-based operating system.
ISIS-IV	—	the 8080/8085-based operating system.
AEDIT	—	an 8086/8088 screen-based text editor.
DEBUG-88	—	a low-level symbolic debugger.
MON85	—	a monitor for 8085 programs.

Related Publications

For more information on the Series IV Microcomputer Development System, see the following manuals:

- *AEDIT Text Editor User's Guide*, 121756;
- *Intellec Microcomputer Development System Overview*, 121752;
- *Series-IV ISIS-IV User's Guide*, 121880
- *DEBUG-88 User's Guide*, 121758
- *NDS-II Network Resource Manager User's Guide Release 3.0*, 135847
- *Intellec® Series IV Installation and Checkout Manual Release 3.0*, 135839

Notational Conventions

UPPERCASE	Characters shown in uppercase must be entered in the order shown. You may enter the characters in uppercase or lowercase.
italic	Italic indicates a meta symbol that may be replaced with an item that fulfills the rules for that symbol. The actual symbol may be any of the following:

directory-name	is that portion of a pathname, that acts as a file locator by identifying the device and/or directory containing the filename,
filename	is a valid name for the part of a pathname that names a file or is a full pathname.
pathname	is a valid designation for a file; in its entirety it consists of a volume name and/or a directory and a filename.
pathname1, *pathname2*, ...	are generic labels placed on sample listings where one or more user-specified pathnames would actually be printed.
system-id	is a generic label placed on sample listings where an operating system-dependent name would actually be printed.
V*x.y*	is a generic label placed on sample listings where the version number of the product that produced the listing would actually be printed.

[ ]	Brackets indicate optional arguments or parameters.
{ }	One and only one of the enclosed entries must be selected unless the field is also surrounded by brackets, in which case it is optional.
{ } . . .	At least one of the enclosed items must be selected unless the field is also surrounded by brackets, in which case it is optional. The items may be used in any order unless otherwise noted.
|	The vertical bar separates options within brackets [ ] or braces { }.
. . .	Ellipses indicate that the preceding argument or parameter may be repeated.
[. . .]	The preceding item may be repeated, but each repetition must be separated by a comma.
punctuation	Punctuation other than ellipses, braces, and brackets must be entered as shown. For example, the punctuation shown in the following command must be entered as shown: `SUBMIT PLM86(PROGA,SRC,'9 SEPT 81')`
input lines	In interactive examples, user input lines are printed in white on black to differentiate them from system output.

<cr>	Indicates a carriage return.
shading	Shading highlights the commands that can only be used if your development system is part of the NDS-II Network (see Chapter 5).

Contents

Contents (Continued)

Contents (Continued)

Tables

Figures

CHAPTER 1 INTRODUCTION

1.1 Introduction

The Intellec Series IV Development System is an 8086/8088- and 8080/8085-based development system. The host execution environment is the iNDX operating system, which allows the Series IV to operate in the 8086/8088 mode. The ISIS-IV operating system, which runs under iNDX, provides the 8080/8085 environment. This manual provides operating and programming information on the iNDX operating system. Figure 1-1 illustrates the Series IV chassis.

1.2 The iNDX Operating System

The Intel Network Distributed Executive (iNDX) operating system resides on hard disk at the Series IV or at the Network Resource Manager (NRM), if part of the Network Development System II (NDS-II). The following paragraphs provide a general description of the operating system.

1.2.1 Human Interface

Interaction with the operating system on the Series IV is through the human interface. There are two features of the human interface immediately apparent to the user: the Command Line Interpreter (CLI), and the Syntax Builder. The Command Line Interpreter provides the keyboard and console interface for the actual entering of commands and command line console display. The Syntax Builder is a software feature that displays iNDX operating system commands,

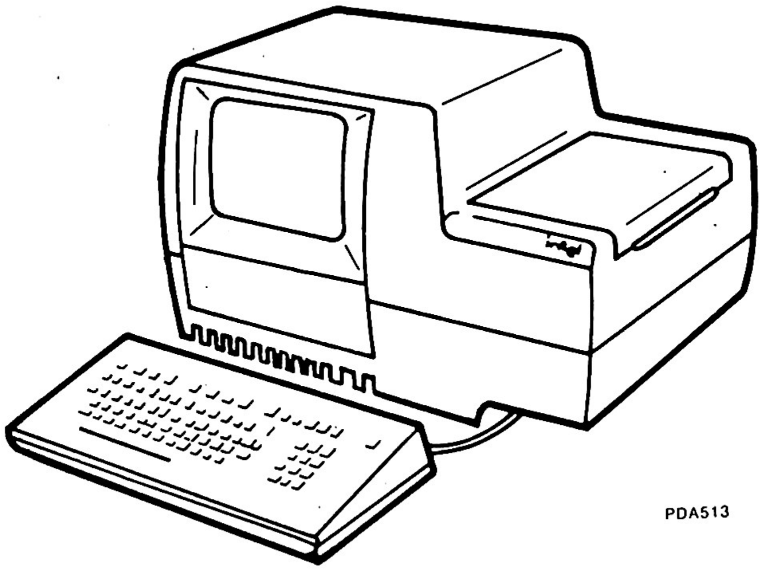


Figure 1-1. The Series IV Development System

command options, prompts, and help files designed to provide assistance in using the operating system. For a more detailed description of these two features of the human interface, refer to Chapter 2.

As seen in Figure 1-2, the human interface also consists of the distributed job control and loader. The command invoked in the CLI will have memory allocated by the Region Controller, then loaded onto system memory by the loader. The distributed job controller identifies and maintains job related information and acts as the queue manager for the Distributed Job Control (DJC) function. The Universal Development Interface (UDI) provides programmatic interface (system calls) between the user programs and the rest of the iNDX operating system.

1.2.2 Distributed File System

The Distributed File System (DFS) provides the interface between the human interface and the rudimentary parts of the operating system (Local File System, BIOS, etc.). Functionally the Distributed File System provides buffered and byte I/O, synchronous interface, file protection and device management. Included in the DFS module are:

- DFS Local Interface—maps UDI system calls and internal interface calls to DFS primitives. It also handles some services; i.e., file protection and user management.
- DFS Network Interface—maps Series IV service requests for remote resources to network protocol; i.e., network LOGON, network file I/O (NDS-II only).

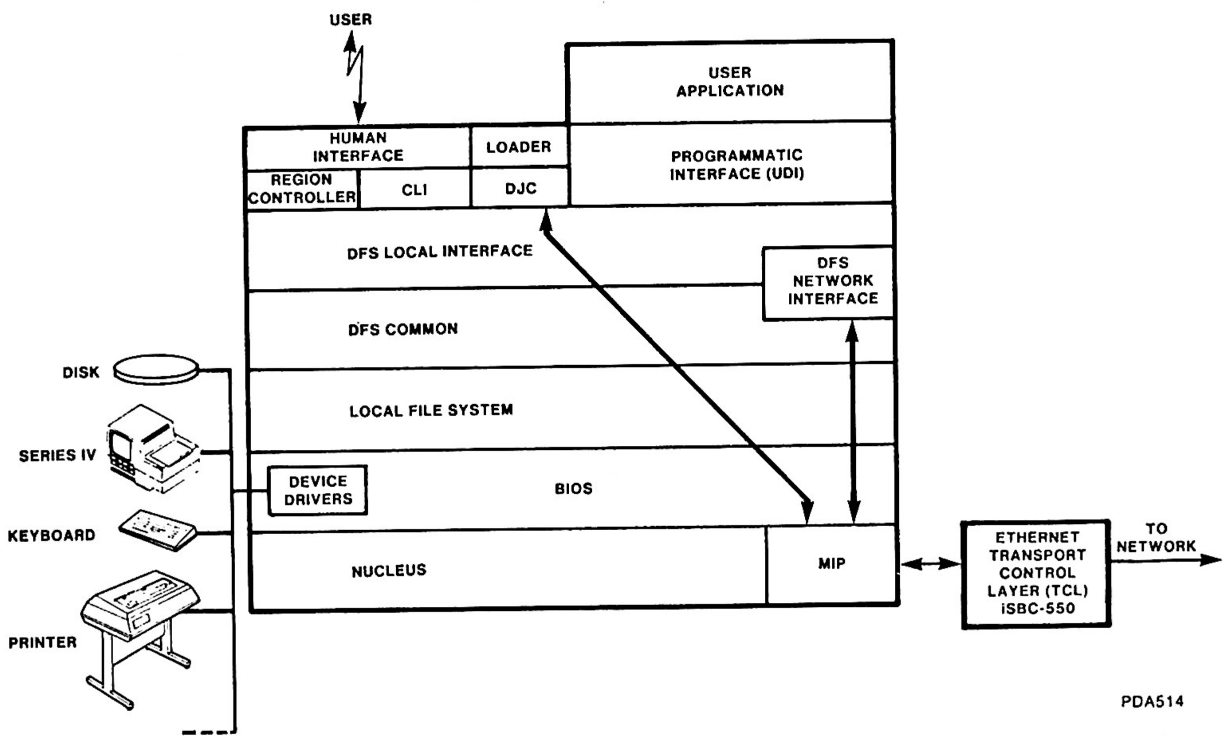


Figure 1-2. iNDX Operating System Block Diagram

- DFS Common—provides the basic distributed file system services to the higher layers of DFS; i.e., buffer management, file I/O, and local device management.

1.2.3 Local File System

The Local File System (LFS) is that portion of the iNDX operating system which provides the basic file functions. Within LFS the file names used in the upper layers of the operating system are converted to physical areas on the disk. The file structure of a disk is kept through the data structures block zero, fnode file, free space map, and bad block map. The fnode file contains essential information on every file on the disk. Pathname components are kept to maintain the directory structure of the disk. All critical system files have duplicate files to maintain device integrity.

1.2.4 Basic Input/Output System (BIOS)

The BIOS in the Series IV contains the device drivers and manager for the mass storage peripheral devices at the Series IV.

1.2.5 The Nucleus

The nucleus of the operating system provides the basic supervisory and management routines that allows the multitasking environment to execute on the Series IV. The dispatcher directs the execution of tasks and the time allowed for execution of each task. A memory manager provides memory allocation for tasks. The interrupt handler within the nucleus is the software interface for the hardware interrupts. Additionally, there is a manager that synchronizes currently executing processes and those waiting for execution. The communication between boards on the Multibus is controlled by the Multibus Interprocessor Protocol (MIP), which is also part of the Nucleus.

1.3 System Initialization

The Series IV development system may be initialized from either the flexible disk or the integrated Winchester disk drive. This section describes the initialization procedures using each device and explains user session termination and system power down.

When the development system is initialized, the power-up diagnostic tests the hardware (internal) power-up sequence.

CAUTION

Always power-up the Series IV before turning on any peripheral devices and turn off any peripherals before shutting down the Series IV.

If you are using a Winchester disk drive as the system device, do not shut the drive off after booting the Series IV.

Neither physically remove nor turn off the system volume after power up.

1.3.1 Booting the Series IV From the Flexible Disk Drive

If your Series IV is configured with both a single flexible disk and an integrated 5¼-inch Winchester disk drive, use the following instructions the first time you boot the operating system. Upon completion of the procedure, perform an iNDX System Build and Cusps Copy to format the Winchester disk and copy the system files from the diskette to the Winchester disk drive. Refer to Chapter 2 of the *Intellec Series IV Installation and Checkout Manual*, for proper procedures. Thereafter, when you initialize the development system, follow the instructions in section 1.3.2.

Refer to Figure 5-1 for the location of switches on the main chassis.

1. Verify that the Configuration switches are set as follows:

   50Hz units—position 7 is ON (up), positions 1-6 and 8 are OFF (down).
   60Hz units—positions 1 and 7 are ON (up), positions 2-6 and 8, OFF (down).

2. Power up the Series IV by turning the MAIN POWER switch located at the left rear of the terminal chassis to position 1 (ON). Wait for the Power-up Diagnostics to complete (disregard any error messages). The following message should be displayed on the CRT upon successful completion of all power-up tests (approx. 40 sec.).

```
SERIES IV SYSTEM POWER-UP DIAGNOSTIC,Vx.y
CPIO PHASE I ...................................... / PASSED
CPIO PHASE II ..................................... / PASSED
FLIPPY DRIVE NOT READY
SPU ............................................... / PASSED
iSBC-550 .......................................... / PASSED
CPIO/SPU MULTIBUS TEST ............................ / PASSED
IEU ............................................. / INSTALLED

SIV Boot Vx.y
Mini-floppy dr.0
-device failure
Error 3800

Attempt to reboot from : Mon88
SERIES IV CPIO MONITOR Vx.y (where x.y is the release level)
```

3. Turn on any peripheral devices, such as a Winchester peripheral chassis or a printer. When using 740 Hard Disk drives, press the STOP/START button and wait for the READY light before proceeding.

4. Insert the 5¼-inch flexible operating system diskette (iNDX.S31, No SPU; iNDX.S41 SPU) (see Figure 1-3) into the right-hand integral disk drive on the Series IV and close the disk drive latch (see Figure 1-4). The read/write access hole should face the rear of the drive and the (uncovered) write enable slot should face the left side of the disk drive.

5. Press the RESET switch (see Figure 1-5) to boot the operating system. The power-up diagnostic test results will appear on the CRT.

6. Verify that the Power-up Diagnostics execute successfully. If they did not, do not proceed with the initialization. Contact your Intel Service Representative.

7. When prompted, enter the appropriate date and time as requested. The format is MM/DD/YY for the date and HH:MM:SS for the time.

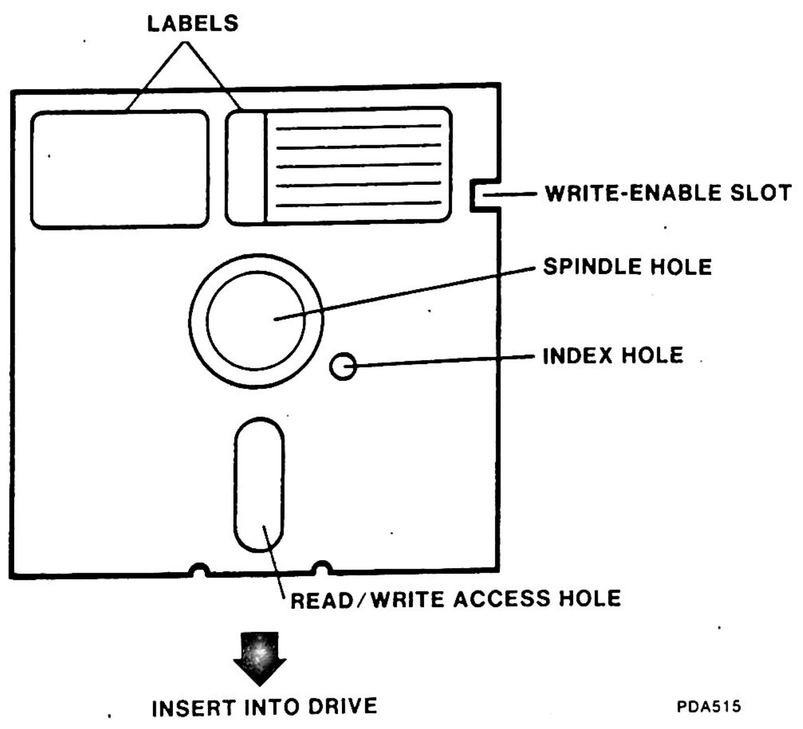


Figure 1-3. Flexible Disk

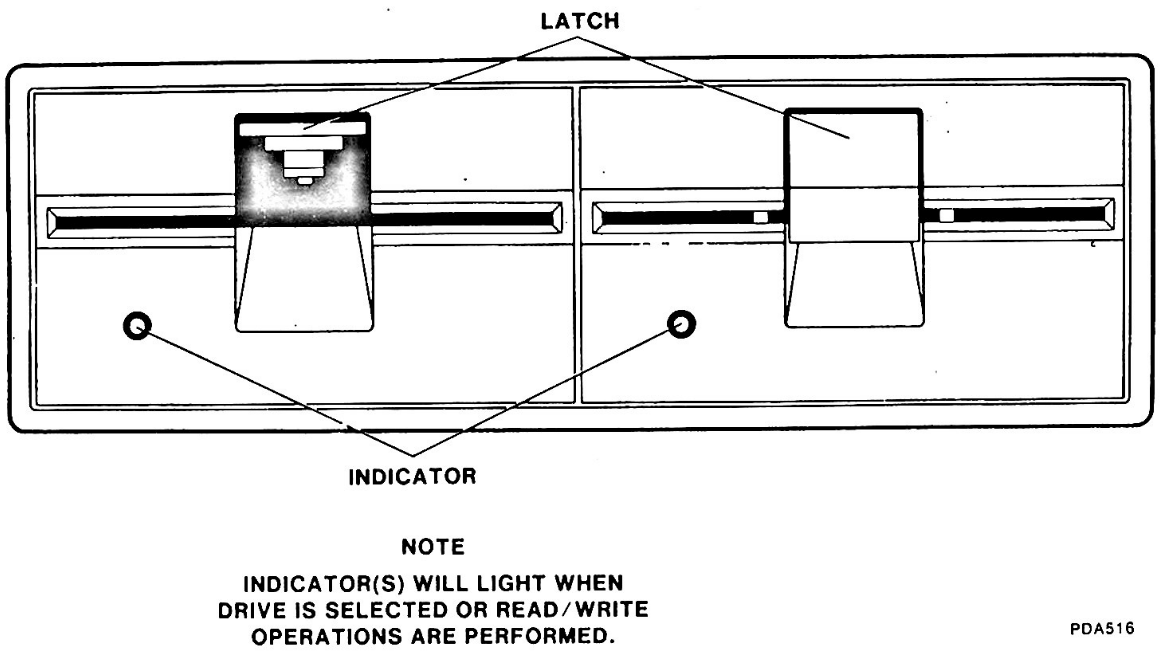

LATCH
INDICATOR
NOTE
INDICATOR(S) WILL LIGHT WHEN DRIVE IS SELECTED OR READ/WRITE OPERATIONS ARE PERFORMED.
PDA516

Figure 1-4. Flexible Disk Drives

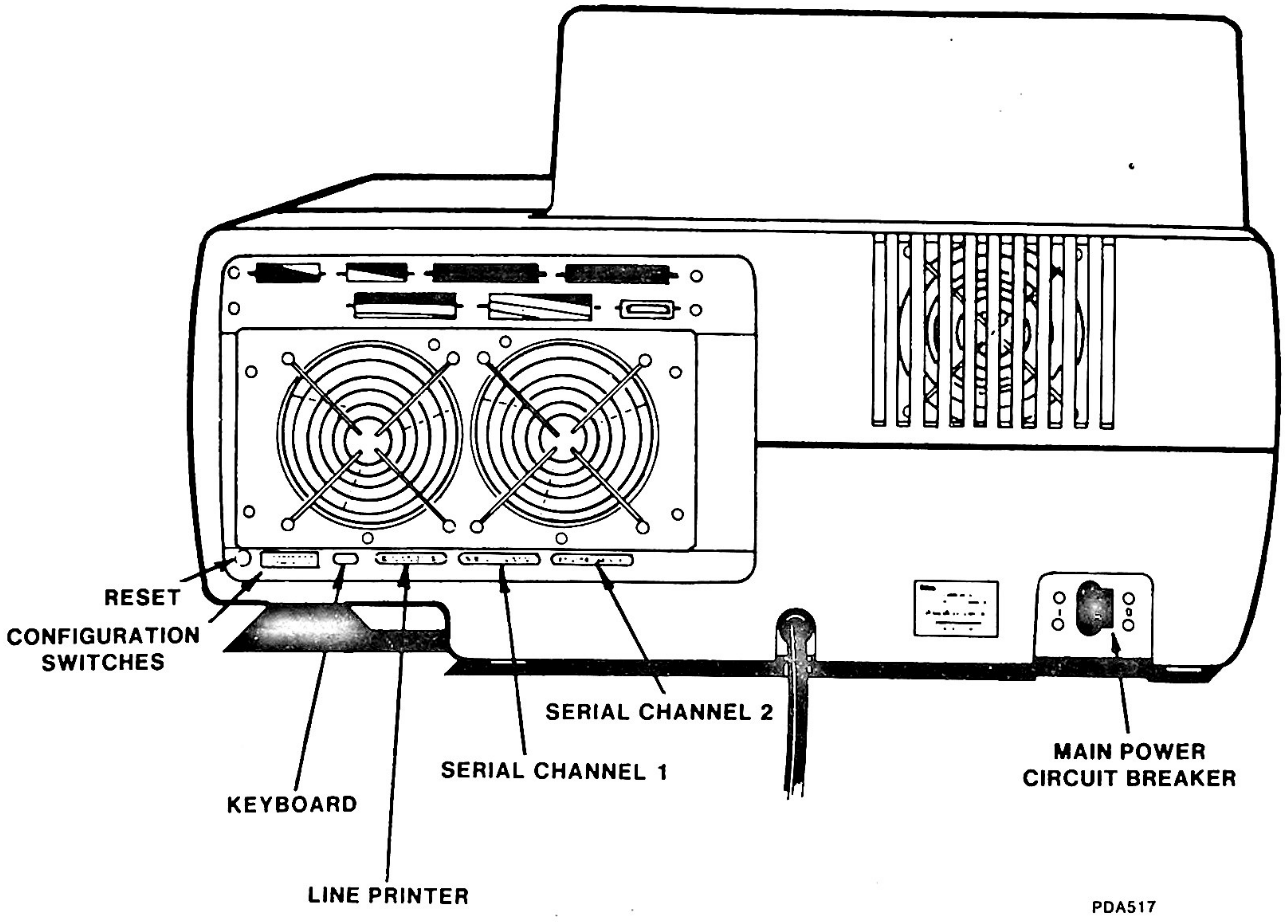


Figure 1-5. Series IV Development System (Rear View)

8. On the system keyboard, type L or F0 and enter your assigned user name. If a Winchester or Hard Disk is part of your system, also enter your password when requested. The first time you log on after installing your system, enter the user name SUPERUSER. Your associated password is PASSME. This is your system identification until you establish other users and passwords. For information on making your files and workstation accessible to others, see Section 3.5 and the USERDEF and CHPASS commands in Chapter 4.

9. Perform the iNDX System Build, Cusps Copy and ISIS build on the Winchester disk drive as instructed in Chapter 2 of the *Intellec Series IV Installation and Checkout Manual.*

The system is now ready to accept commands. Use the facilities detailed in the *Series IV Microcomputer Development System Overview Manual*, to assist you in learning to interface with your system.

1.3.2 Booting the Series IV From the Integrated Winchester Drive

To boot the Series IV from the integrated 5¼-inch Winchester disk drive after doing an iNDX System Build, Cusps Copy and ISIS build, follow these steps:

1. Verify that the configuration switches (see Figure 1-5) are set as follows:

   for 60Hz operation: positions 1, 5, and 7 are ON (up)
   positions 2-4, 6, and 8 are OFF (down)

   for 50Hz operation: positions 5 and 7 are ON (up)
   positions 1-4, 6, and 8 are OFF (down)

2. Power up the Series IV by turning the MAIN POWER switch located at the left rear of the terminal chassis to position 1 (ON).

3. Turn on any peripherals such as a Winchester peripheral chassis or a printer. When using 740 Hard Disk drives, press the STOP/START button and wait for the READY light before proceeding.

4. Press the RESET switch (see Figure 1-5) to boot the operating system. The Power-up Diagnostic results will appear on the CRT

5. Verify that the Power-up Diagnostic tests execute successfully. If they do not, do not proceed with the initialization. Contact your Intel Service Representative.

6. When prompted, enter the appropriate date and time as requested. The format is MM/DD/YY for the date and HH:MM:SS for the time.

7. At the system keyboard type L (logon) or F0 and enter your assigned user name, then your password.

   The first time you log on after installing your system, enter the name SUPERUSER. Your associated password is PASSME. This is your system identification until you establish other users and passwords. For information on making your files and workstation accessible to others, see "File Access and Ownership" in Chapter 3 and the USERDEF and CHPASS commands in Chapter 4.

The system is now ready to accept commands.

1.3.3 Booting the Series IV From the External Winchester Drive

To boot the Series IV from the external 8-inch Winchester disk drive after doing an iNDX System Build, Cusps Copy and ISIS Build, perform the following steps:

1. Verify that the configuration switches (see Figure 1-5) are set as follows:

   for 60Hz operation: positions 1, 6, and 7 are ON (up)
   positions 2-5, and 8 are OFF (down)

   for 50Hz operation: positions 6 and 7 are ON (up)
   positions 1-5, and 8 are OFF (down)

2. Power up the Series IV by turning the MAIN Power switch located at the left rear of the terminal chassis to position 1 (ON).

3. Turn on the External Peripheral Chassis by setting the power switch located on the front panel to position ON.

4. Turn on any other peripherals connected to the Series IV. When using 740 Hard Disk drives, press the STOP/START button and wait for the READY light to turn on before proceeding.

5. Press the Series IV RESET switch (see Figure 1-5) to boot the operating system. The power-up diagnostic results will appear on the CRT.
6. Verify that the power-up diagnostic tests execute successfully. If they do not, do not proceed with the initialization. Contact your Intel Service Representative.
7. When prompted, enter the appropriate date and time as requested. The format is MM/DD/YY for the date and HH:MM:SS for the time.
8. At the system keyboard type L (logon) or F0 and enter your assigned user name, then your password.

   The first time you log on after installing your system, enter the name SUPERUSER. Your associated password is PASSME. This is your system identification until you establish other users and passwords. For information on making your files and workstation accessible to others, see "File Access and Ownership" in Chapter 3 and the USERDEF and CHPASS commands in Chapter 4.

1.3.4 Terminating a User Session

To terminate the user session on the Series IV and allow another user to start a session, follow these steps:

1. Log off by entering the command LOGOff or by pressing the appropriate function key (see Figure 2-1).
2. The new user may now log on by entering his/her user name and a password.

NOTE

To terminate the multi-user mode enter LOGOff Exit.

1.3.5 Powering Down the Series IV

To power down the Series IV development system, follow these steps:

1. Log off by entering the command LOGOff or by pressing the appropriate function key (see Figure 2-1).
2. Turn off any peripheral devices.
3. If you are using a flexible disk(s), wait for the drive indicator light to go off (see Figure 1-4). Release the drive door latch(es) and remove the flexible disk(s).
4. Move the MAIN POWER switch located at the left rear of the terminal chassis to the 0 (OFF) position.

1.4 Disk File Types

A disk is either a system disk or non-system disk, depending on its iNDX files.

- A system disk contains the files necessary to boot the operating system.
- A non-system disk contains only the files necessary for the creation and storage of files, leaving more space for data than on a system disk.

When the system is reset with an iNDX system disk in the appropriate drive, the operating system initializes and takes control of the system.

At system start-up, only the essential iNDX files are loaded into memory. iNDX command files remain on disk until a command is entered that calls them. The required program is then loaded into memory and executed. After the command program has completed its functions, the memory it was using is again available. This allows efficient use of the operating system and memory.

The basic types of files are:

- iNDX system structure files—contain the information necessary to maintain the file system structure on each disk.
- iNDX system files—contain both the basic system programs and the command programs (data files)
- User created files (data and program).
- Directory files.

CHAPTER 2
HUMAN INTERFACE

2.1 Introduction

The human interface module of the Series IV provides a command language interpreter (CLI) that processes command input, initiates command execution and, when execution is complete, prompts for another command. A batch command processing facility is available through the SUBMIT, BACKGROUND, BATCH, and EXPORT commands to allow the execution of a series of commands stored in a file. CLI also provides line editing facilities.

The terminal session is initiated by the iNDX operating system asking the user to log on. When log on is completed, the operating system is in Command Mode (the prompt {>} is displayed). At this time, the user may enter any valid command sequence. To terminate the session, it is necessary to log off. To terminate execution of a command or submit file, use the BREAK key or CONTROL-C on the terminal keyboard.

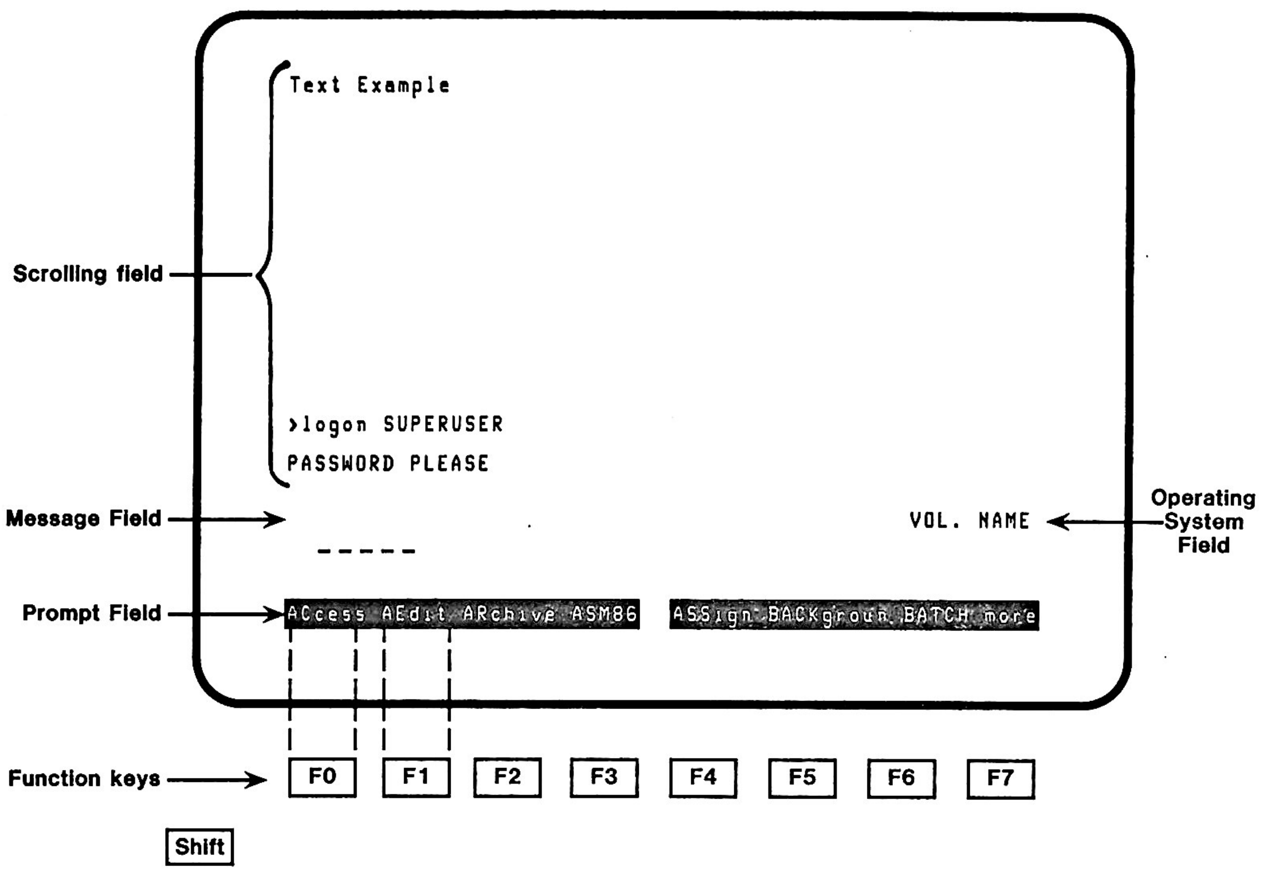
2.2 The Intellec Terminal

The Intellec terminal is comprised of the display screen and the keyboard.

2.2.1 Display Screen

The display screen (shown in Figure 2-1) is partitioned into four display fields. Information entered in a given field never carries over into another field.

Scrolling Field:	Lines 1–23 form the Scrolling Field. Each line is 80 characters wide. User-entered text will appear in the Scrolling Field. If more than 23 lines are entered, the field scrolls up one line. The topmost line disappears, additional data entries appear on Line 23.
Message Field:	The Message Field occupies character positions 1–60 of line 24. Many system programs use the Message Field as a display area to avoid disturbing text that appears in the Scrolling Field.
Operating System Field:	Character positions 61–80 of line 24 form the Operating System Field. The operating system uses this field to display Job Control messages, the names of mounted volumes, and the names of background and remote jobs.
Prompt Field:	The Prompt Field occupies line 25. This field is used by the Syntax Guide to display Menu entries command options. A maximum of 8 entries may appear on line 25 consisting of two reverse video blocks of four entries (commands or options). Each of the entries is associated with one of the 8 Function keys.



```
Text Example

>logon SUPERUSER
PASSWORD PLEASE
                                                VOL. NAME
 -----
ACcess AEdit ARchive ASM86   ASSign BACKgroun BATCH more
```

Figure 2-1. The Display Screen

2.2.2 Character Display

Characters in the display fields will appear on the screen overlined, in reverse video, blinking or highlighted.

The cursor appears on the screen as a nonblinking, reverse video rectangle. The cursor moves 1 character position to the right with each keystroke until it reaches Column 80 (the right-most column). Pressing the RETURN key causes the cursor to move to the initial (left-most) character position of the next line. If the cursor is on the bottom-most line of text (i.e., Line 23 of the Scrolling Field), the text is scrolled up one line.

2.2.3 The Keyboard

The keyboard, shown in Figure 2-2, is the user's input interface with the system. From the keyboard it is possible to control the system, enter data and commands, and request data. The data entered at the keyboard is stored in a line editing buffer until the RETURN key is depressed.

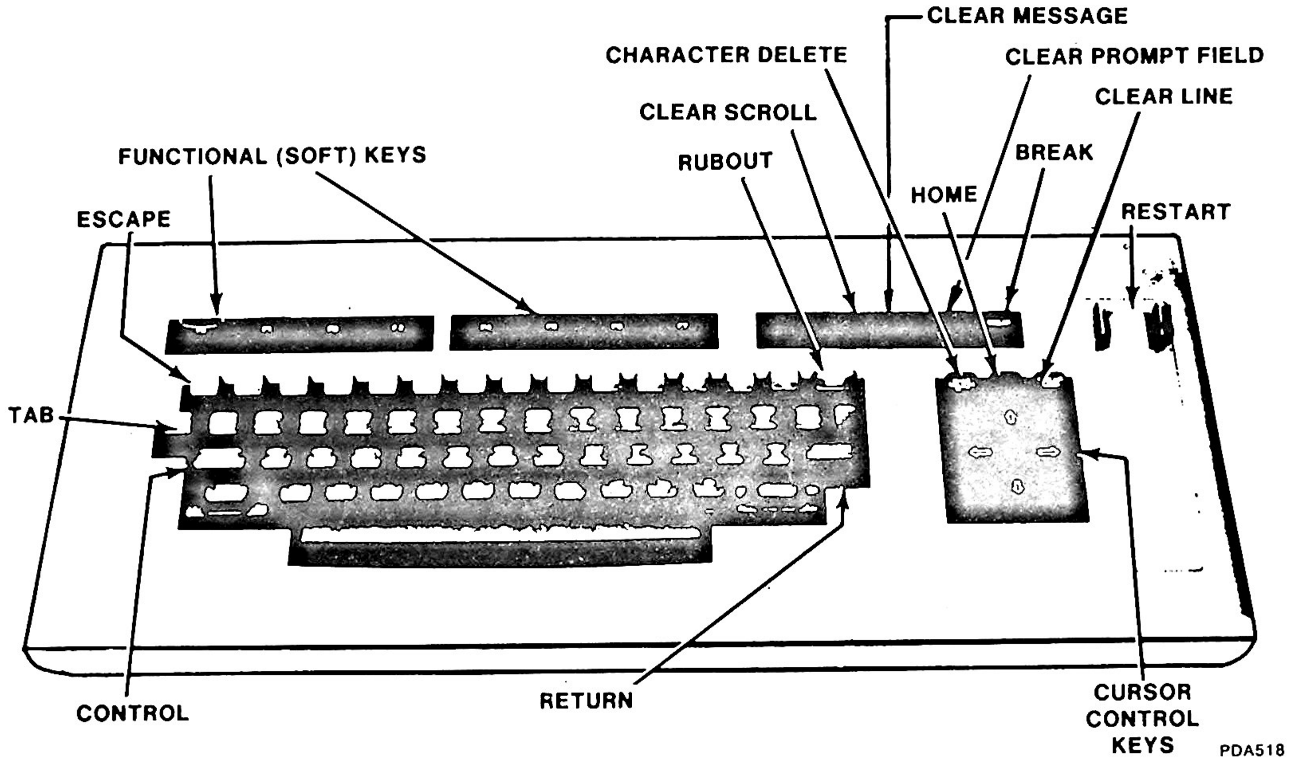


Figure 2-2. Series IV Keyboard

2.2.3.1 Key Clusters

The keyboard is organized into four clusters of keys: function keys, alphanumeric keys, reserved keys, and edit keys. Following is a description of each key cluster and several special-purpose keys.

Function Keys:	When the Command Line Interpreter is executing, these keys are used in conjunction with the Menu items in the Prompt Field (line 25 of the Display Screen). Each of the eight Function keys is associated, position-by-position, with one of the Menu items. To select a given Menu item, press the key associated with that item. For example, to select the third Menu item (counting from left-to-right), press the key inscribed F2 (F0, F1, F2 being the first three keys from left-to-right). Pressing a Function key while holding down the SHIFT key produces the Help Text for the Menu entry associated with that key.
Alphanumeric Keys:	Each of these keys generate the character inscribed on the keycap.
Reserved Keys:	The cluster of five keys located in the upper right of the keyboard are reserved for future system use.

Edit Keys:	The Edit keys consist of the keypad on the right of the keyboard and several other special keys along the right and left edges of the alphanumeric key cluster. Of the eleven keys in the right keypad cluster, only seven have inscribed keycaps. The remaining keys do not have assigned functions. The line editing keys are described in Table 2-1.
SHIFT Key:	If two characters are inscribed on a given alphanumeric key, the SHIFT key must be pressed with the alphanumeric key to produce the character inscribed on the upper part of the keycap. When the SHIFT key is not used, the lower character is generated. Pressing the SHIFT key with an alphabetic key produces an uppercase character.
CAPS LOCK Key:	The CAPS LOCK key functions only with the twenty-six alphabetic keys. It allows the user to enter a series of uppercase alphabetic letters without having to hold down the SHIFT key. To enter a string of uppercase characters, press the CAPS LOCK key to its lower, "locked-in" position. To generate lowercase characters again, press the CAPS LOCK key again, returning it to its upper, "unlocked" position.
BREAK Key:	The BREAK key, located to the right of the reserved key cluster, aborts the execution of a job and returns the system to the interactive level (foreground).
RESTART Key:	This key is reserved for use by field service personnel.

Table 2-1. Line Editor Features

Key Name	Function
RETURN	1. Terminates the line at the current cursor position. 2. Enters the command line into the system.
ESCAPE (ESC)	1. When entered as the first character in a command line, recalls the last line to the display. 2. Terminates the line at the right margin, not at the current cursor position (as with RETURN).
RUBOUT	Deletes the character to the left of the cursor and moves the cursor left one position.
CTRL X (CONTROL + X)	Deletes all characters in the current line which are to the left of the cursor. The remainder of the line is re-displayed (left-justified) with the cursor at the left margin of the line.

Table 2-1. Line Editor Features (Cont'd)

Key Name	Function
CTRL A (CONTROL + A)	Deletes all characters from the current cursor position to the end-of-line. The cursor position does not change.
DEL CHAR	Deletes the character at the cursor location. The ursor position does not change.
CLEAR LINE	Deletes the entire line. The cursor position does not change. Control remains in the line editor.
↑ (up arrow)	Moves the cursor up one line, but retains column positioning. Not functional in Command Mode.
↓ (down arrow)	Moves the cursor down one line, but retains column positioning. Not functional in Command Mode.
→ (right arrow)	Moves the cursor one position to the right but not past the current end-of-line.
← (left arrow)	Moves the cursor one position to the left but not past the starting position.
HOME	Moves the cursor position to the current end-of-line. If the last character entered was a left arrow, this key moves the cursor to the starting position.
CTRL S (CONTROL + S)	Stops output to the console.
CTRL Q (CONTROL + Q)	Resumes output to the console.

2.3 Console Operation

The iNDX operating system provides two features that help the user enter commands: the Syntax Guide and the Help Text. The Syntax Guide presents each command and its options as Menu entries. Chapter 4 provides detailed descriptions and use of the commands, and should be consulted frequently during initial system use.

The Help Text provides a functional description of a given command and its elements and options. To obtain the appropriate Help Text simply press the Function key (F0-F7) associated with that element while holding down the SHIFT key.

2.3.1 Entering Commands—Interactive Mode

The Syntax Guide displays commands and options as Menu entries in the Prompt Field (line 25 of the Display Screen). Up to eight entries, grouped into two blocks of four entries each, can be presented at one time. This arrangement emphasizes the correspondence between the Menu entries and the eight Function keys. To select a given Menu entry, either enter it from the keyboard or press the Function key spatially associated with that Menu entry. The first (left-most) Menu entry is associated with the F0 key, the second with the F1 key, and so on, to the eighth (right-most) entry, which is associated with the F7 key. Usually, the eighth Menu entry is "--more--", indicating that additional Menu entries can be displayed by pressing the F7 or TAB keys. The Menu scrolls, eventually returning to the initial display. If the keyboard entry method is used rather than the Function key method, it is necessary to know if the FILL option is operative (see the description of the FILL command in Chapter 4). If FILL is operative enter only the letters shown in uppercase in the Menu line; the system fills in the remainder of the command automatically. If FILL is inoperative, you must enter the complete command.

The available commands and the associated function keys are as follows:

```
ACcess  AEdit   ARchive  ASM86    ASSign  BACkgrou-  BATch      -more-
CAncel  CC86    CHOwner  CHPass   COPy    COUnt      CREATedir  -more-
CREF86  DElete  DIR      DISmount ELse    END        ENDJob     -more-
EXIt    EXPort  FIll     FORMat   FORT86  I2ice      ICopy      -more-
IF      IMport  ISis     LIB86    LINK86  LName      LOC86      -more-
LOG     LOGOff  MAIl     MAKe     MOunt   OH86       OPen       -more-
ORif    PAsc86  PDscopy  PLm86    PScope  Queue      REAd       -more-
REGion  RELab   REName   REPeat   RUn     S4fprt     SDcopy     -more-
SEArch  SET     SPace    STty     SUbmit  SVcs       SYSGen     -more-
SYSTat  TIme    UNtil    USERDef  USERS   VErify     VIew       -more-
WAit    WHIle                                                   -more-

 F0      F1      F2       F3       F4      F5         F6         F7
```

For each Menu entry, the accompanying Help Text can be obtained by holding down the SHIFT key while pressing the appropriate Function Key. For example, consider the following eight Menu entries:

```
CREF86  DElete  DIR  DISmount     ELSE  END  ENDJob  -more-

 F0      F1      F2   F3           F4    F5   F6      F7
```

To select the DELETE command, press the F2 key or, assuming for the example that FILL is operative, enter the characters DE. (If FILL was inoperative, enter DELETE). To obtain Help (more information) about the DELETE command, press the SHIFT and F2 keys simultaneously.

Once DELETE has been selected from the Menu line (either by pressing the Function key or by entering the command from the keyboard), the Menu line will no longer display the original eight entries. Instead, the Menu line will now contain the following message:

```
ENTER <file name>
```

where *file name* is the file to be deleted. Suppose an entry request is not clearly understood. Since the original Menu line is not on the screen, Help cannot be obtained by pressing SHIFT and a Function key. The Syntax Guide allows the user to "back up", thus recalling the previous Menu line. To back up, press the cursor left (←) key. The original Menu line will now reappear. To obtain more information about the DELETE command, press the Function key associated with the DELETE Menu entry while holding down the SHIFT key.

After reading the Help Text, return to the request for a file name by pressing the cursor right (→) key. As long as valid characters are entered, the Menu line will remain the same. The file name is terminated with a delimiter by pressing the space bar. Note that the Menu line has changed again and is now listing the options for the DELETE command, as shown below:

```
DIR QUERY --exec--
```

Select either (or both) of the options by pressing the Function key associated with them (i.e., F0 and F1), or by entering D or Q at the keyboard (DIR or QUERY if FILL is inoperative). If more information about the two options is desired, press both the SHIFT key and Function key associated with the desired option. If a command line option is selected, but more option information is desired, use the left arrow key (←) to move the cursor back to the 1st letter of the option. The function key and shift can then be used to display the help files for command option.

After entering the entire command line, execute the command by pressing the Function key associated with the "--exec--" entry or by pressing the RETURN or ESCAPE key.

2.4 Command Line Interpreter

The Command Line Interpreter (CLI) is a program with direct user interaction. The CLI requests input with the > prompt, accepts a complete command from the input device, substitutes the CLI variables with the defined values, loads, and invokes the requested program. A complete command may require several input lines, each with a maximum of 128 characters. Each successive input line then, except the last, must end with the continuation character, an ampersand. When the CLI detects the ampersand, it issues a continuation prompt (> >) and allows continuation of the command entry on the next physical line. Only when a complete command line has been entered (with a carriage return <cr>) is the program loaded and invoked. If comments are to be included in a command line, use a semicolon (;). All characters between the semicolon and the return are recognized as a comment and are ignored in execution.

2.5 Command Delimiters

A command consists of a sequence of characters. When a command is examined by the system prior to execution, special characters called delimiters are used to divide the command into words that are treated as units. The following characters function as delimiters:

```
 !      <
 #      =
 $      >
 %      [
 (      \
 )      ]
 +      ‘
 ,      |
 -      space
<cr>
```

A delimiter can be used within a string by enclosing the string within quotation marks.

Pathnames must not be broken by an ampersand. If a pathname is entered as part of a command and the pathname contains a delimiter, enclose the pathname in quotation marks so the system will treat it as a unit. For example, if the pathname /VOL1.A/A!B is entered as part of a command, enclose it within quotes:

```
COPY "VOL1.A/A!B" TO /VOL2.B
```

2.6 Command Line Input

The terminal supports a "type ahead" operation. This means that up to thirty-two keystrokes may be saved in the buffer until the system is ready for them. When the buffer is full, the system issues an audible beep; additional characters are discarded.

The command line is not complete until a RETURN or ESC key is used to terminate the line. During line input, the cursor (a reverse video block) indicates the position of the next character. In line edit, the cursor does not move off the current line or to the left of the starting position (the third character position). If more characters than can be displayed on the current screen line are entered, an exclamation mark (!) and the cursor will appear in column eighty.

2.7 CLI Variables

CLI variables are symbols that have string values associated with them. The command language allows these variables to be defined and referenced within a command file. The scope of CLI variables is restricted to the command file in which the CLI variables are defined. The CLI variable name can be a maximum of six alphanumeric characters. The first character of the name must be alphabetic. Letter case is not significant. The value associated with a CLI variable is a string with a maximum of five hundred and eight ASCII characters. A reference to a CLI variable consists of a percent sign (%) followed by the name of the CLI variable.

The CLI must be able to distinguish the variable reference from the surrounding text. One of the following techniques can be used:

1. Follow the variable name with a delimiter character.
2. Enclose the variable reference in matching quotes. The quotes are removed if the program obtains its arguments through the DQ$GET$ARGUMENT$ function (see Chapter 7).
3. Make the variable name a full six characters long. The CLI stops looking if a delimiter has not been encountered within six characters.

When a reference to a CLI variable is encountered in a command line read by CLI, the interpreter replaces the reference with the current value of the CLI variable before command execution. If the referenced CLI variable is undefined, the reference becomes a null string. Examples of CLI variable use appear in Figure 2-3.

There are two different types of CLI variables: system-defined and user-defined. No more that ten CLI variables can be defined within one command file, including both system-defined and user-defined variables.

2.7.1 System-Defined CLI Variable

One system-defined CLI variable is provided in the Series IV: 'STATUS'. Any reference to it takes the form '%STATUS'. At any given point, the value of STATUS represents the completion code returned by the last DQ$EXIT call executed (see Chapter 7). The completion code is converted to a string of ASCII decimal digits, thereby expressing the value of the completion code in decimal notation (leading zeros are suppressed). Control structures, together with the STATUS

```
Command File (compil.csd)
      OPEN FILES.NRM                 ; get parameter file
      SET LINKEM TO ' '                    ; initialize
                                           ; string to
                                           ; hold link file
                                           ; name
      REPEAT
           READ NAME                       ; get module name
IF %STATUS = 0 THEN
           WHILE %NAME < > ' '             ; exit loop at end of
                                           ; parameter file
           ;
           ;   compile one module
           ;
           PLM86 %NAME.P86 DEBUG COMPACT
           ;
           ;   build string of file names for link step
           ;
           IF %LINKEM = ' '
                SET LINKEM TO '%NAME.OBJ'
           ELSE
                SET LINKEM TO '%LINKEM,%NAME.OBJ'
           END
           END
      END
      ;
      LINK86 %LINKEM, :F0:SYSTEM.LIB TO DRIVER BIND
Parameter File (files.nrm)
      driver, mod1, mod2
Invocation Line
SUBMIT compil
```

Figure 2-3. Command File Example

variable, make possible the conditional execution of subsequent steps in a command file.

2.7.2 User-Defined CLI Variables

CLI variables may also be created by the user by means of the SET and READ commands (see Chapter 4).

2.7.3 Commands for Manipulating CLI Variables

The SET command is an assignment statement for CLI variables. If the receiving variable does not exist, SET creates it. The string concatenation of any combination of literal strings and the values of CLI variables can be assigned to a CLI variable (either system-defined or user-defined), as shown in the following example.

Assuming FILE and EXIT are user-defined CLI variables with the following values:

%FILE = "aprog"
%EXT = "obj"

The statement:

```
SET NAME TO ``%FILE.%EXT''
```

results in:

%NAME = "aprog.obj"

In this example, the quotes surrounding the values of the CLI variables are not part of the values, but are included to separate the string values from the surrounding text.

As CLI processes a command line, the following sequence occurs: (1) the line is read, (2) the line is scanned for references to CLI variables and all substitutions are performed, (3) the line is parsed as a command. The literal quotes around the object following the TO are necessary if the values of the CLI variables being substituted contain delimiters.

2.7.4 Examples of CLI Variable Substitution

Assume the following CLI variables have been defined:

%FILE = "mod"
%NUMBER = "1"
%EXT = "p86"

Consider the following command lines before substitution, after substitution, and at execution (i.e., the command tail as it appears when retrieved by the DQ$GET$ARGUMENT UDI primitive).

```
before:    plm86 %FILE%NUMBER.%EXT
subst:     plm86 mod1.p86
exec:      plm86 mod1.p86

before:    plm86 "%FILE"A%NUMBER.%EXT
subst:     plm86 "mod"A1.p86
exec:      plm86 modA1.p86
```

The quotes are necessary because the CLI would interpret "%FILEA-%NUMBER" as a command line containing a reference to a CLI variable called "FILEA".

```
before:    plm86 %FILE%NUMBERA.%EXT
subst:     plm86 mod1A.p86
exec:      plm86 mod1A.p86
```

Quotes are not necessary in this example because the CLI knows that variable names are a maximum of six characters. CLI assumes the second variable reference is to "NUMBER" rather than to "NUMBERA".

If a literal percent sign needs to be included in a command line it can be surrounded by matching quotes:

```
before:     a"%"FILE.obj
subst:      a"%"FILE.obj
exec:       a%FILE.obj
```

2.8 Command Files

Commands are normally entered from the console. However, permanent files that contain lists of commands can be created and maintained. These permanent files are called command files. Command files can be executed in the foreground of the system by using the SUBMIT command (see Chapter 4), in the background of the system by using the BACKGROUND command (see Chapter 4), or at a remote workstation (NDS-II only) by using the EXPORT command (see Chapter 5).

Refer to Figure 2-3 for an example of command file usage. In addition to the normal console commands, the Series IV has control commands that provide, at run time, conditional or repetitive execution of a set of commands within a command file. CLI variables may also be defined within the command file.

2.8.1 Dynamic Command File Creation

When a command is received by the CLI from the keyboard, the syntax builder is invoked. Prompts are always displayed, although you can disable command keyword completion (refer to the FILL command in Chapter 4). If you attempt to create a command file with a standard text editor, the syntax builder prompts will not occur. If you were dependent on these, you might have to check a manual or the corresponding reference card to create a command file. To avoid this inconvenience, CLI provides the BATCH command. The BATCH command uses the syntax builder to create, write, and execute a command file. Another advantage of the BATCH command is that, since it is part of the CLI, less memory is required to invoke it than is required to invoke a separate text editor. When the BATCH command is invoked, only the keyword BATCH and the pathname of the command file is specified.

If the command file already exists, the syntax builder's editor can be used to alter the contents of the command file. If the file does not exist, it is created and then written at the end of the edit process. When the command file is complete, the BATCH command prompts the user to select one of the following options: abort, write it without executing it, execute it in the foreground or background, or export it to a remote station for execution. BATCH allows the user to write and execute the file, or execute it only (a temporary file is created and then deleted after execution). If the executed file contains references to formal parameters, BATCH prompts for the actual parameter values to be used. Refer to Chapter 4 for a detailed description of the syntax and operation of the BATCH command.

2.9 Log Files

The Human Interface provides a log facility that allows the user to specify that output written to the logical console output device should also be written to a mass-storage file. In a foreground job, normal console output is displayed on the physical screen. If a log file is active, the same output is also written to the log file. In a background job, the normal console output goes to the Byte Bucket (see Section 2.12). A log file can be requested in either of two ways: by executing the LOG command, or by specifying the LOG option of the SUBMIT, BACKGROUND, and EXPORT commands with the LOG option (default). (The LOG command is valid only from the keyboard. See Chapter 4 for a description of the LOG command.)

The log file is the only attribute of a command file environment which is inherited by nested command files. If a log file is active when a SUBMIT command is issued, the console output from the newly submitted command file is also written to the currently active log file unless the LOG keyword is specified on the SUBMIT command. If a log file is active when a SUBMIT command with the LOG keyword is issued, the current log file is detached before the new log file is created. When the inner nested command file has finished executing, its log file is detached, the log file of the outer command environment is re-attached, and the file pointer is positioned at the end-of-file. The log file of the outer command environment (either the keyboard or another command file) resumes with the next commmand after the SUBMIT (with LOG keyword).

2.10 Parameter Substitution

Actual parameters can be specifed when a command file is submitted for execution. The command file can have formal parameters of the form %n, where n is a decimal digit (0-9). At submit time, a list of actual parameters is supplied along with the name of the command file to be executed. Before the command file is executed, the CLI creates a new copy of the command file where all occurrences of formal parameters have been replaced by the corresponding actual parameters. The formal parameter, %n, is replaced by the (n + 1)st element of the list of actual parameters (%0 is replaced by the first list element, etc.). The parameter replacement is done by scanning each line in the command file once from left-to-right. Every occurrence of the string, %n, is replaced by the corresponding actual parameter in a string substitution. If the actual parameter is enclosed in quotes, the quotes are removed before the substitution is performed. If a formal parameter has no corresponding actual parameter, the replacement is performed using a null string.

2.10.1 Parameter Files

To increase the power of command files as utility tools, the command language has commands that allow the values of CLI variables to be read from mass-storage files. These files, referred to as parameter files, are manipulated using the OPEN and READ commands. Only one parameter file can be open at any given time. The OPEN statement allows any pathname to be specified as a parameter file. The READ command treats the parameter file as a byte stream subdivided into strings by delimiters, and specifies a list of CLI variable names. The READ command proceeds from the current file pointer position in the parameter file and assigns a string to each variable in the list. If end-of-file is detected and the STATUS variable is set to a non-zero value on the parameter file during a READ, the parameter file is closed. If the list contains more variables than the number of strings in the file, the variables without corresponding strings are set to null.

Suppose file "files.nrm" contains the following two logical lines:

drive, mod1
mod2, mod3

The execution of the command file fragment

```
OPEN FILES.NRM
READ FILE1, FILE2, FILE3
```

results in values of the CLI variables FILE1, FILE2, and FILE3:

%FILE1 = 'drive'
%FILE2 = 'mod1'
%FILE3 = 'mod2'

The quotes are not part of the values of the variables. If a subsequent READ command is issued, the string retrieved is 'mod3'. Thus, general purpose utility command files that can process groups of related modules, whose names are specified by parameter files, can be constructed. An example for such a command file is given in Figure 2-3.

2.11 System-Designated Device Names

The following device names are defined by the operating system:

:TI: or :TI1:	Serial channel #1 input
:TO: or :TO1:	Serial channel #1 output
:TI2:	Serial channel #2 input
:TO2:	Serial channel #2 output
:LP:	Line printer (local)
:SP:	Spool printer
:CI:	Console input (typically, Series-IV keyboard in foreground)
:CO:	Console output (typically, Series-IV display in foreground)
:BB:	Byte Bucket Though nonexistent, the byte bucket is treated as a real device by the commands. The byte bucket receives data you want to discard. Writing to :BB:, always successful, simply discards data. Reading from :BB: returns an end-of-file (i.e., a zero byte read).
FL0, FL1	Flexible disks
WM0 – WM3	Integrated 5¼-inch Winchester disk
WD0 – WD3	Winchester 35 Mb disk
HD0 – HD3	HD 5440 hard disks

2.12 Modes of Operation

The iNDX operating system has three modes of operation that allow varying degrees of interaction to the multi-tasking capabilities. There are two regions or memory partitions available where independent tasks are executed. The default mode is single-user which maintains a background region that may be used to process non-interactive batch jobs. When a second terminal is attached to serial channel 1, the Series IV may be operated in multi-user mode. This mode maintains two entirely independent regions—one for the Series IV console and one for the attached terminal. The third mode is known as toggle mode, and allows a single user to access and utilize the two regions interactively.

2.12.1 Single User Mode

In single-user mode, the foreground region is used and accessed by the user interactively and a second or background region is used to execute non-interactive command files with the Background command. A background job runs in the batch mode in the background concurrently with the foreground job. Syntactically, the BACKGROUND command is similar to the SUBMIT command: you specify the name of a file that contains a sequence of commands to be executed. When the BACKGROUND command is executed, job control creates the background job environment, and logs on as a background job. When the command file has been exhausted, job control logs off and deletes the background job. Once created, a background job has no relationship to the foreground job from which it was created; nor does it inherit any environmental information (e.g., logical names or CLI variables) from the foreground job, but does take environmental information from the INIT file.

The background job has no physical console attached to it. Thus, certain programs cannot be executed in the background and certain primitives are disallowed. A screen-oriented editor or debugger should not be invoked from the background. If a program running in the background calls DQ$TRAP$CC (see Chapter 7), the call returns the message, EXCEPT = E$OK, but no action is taken. If CONTROL-C is typed, the CONTROL-C handler of the foreground job is always invoked. A background job ignores a CONTROL-C request but the BREAK key can be used to terminate a background job.

A useful feature of the single-user mode is that with the STTY REMOTE option, the Series IV may be controlled with a second or remote terminal. The primary console is inactivated at this time. This feature may be useful to the developer with a terminal at home.

Another feature that may be used in the single-user mode is STTY TERMINAL. This option allows the Series IV to act as a terminal that may be connected to a host computer. The interface is programmable, using the configuration file as documented in Appendix D.

2.12.2 Multi-user Mode

To use the multi-user mode, a terminal must be attached to serial channel 1 or serial channel 2 and the terminal characteristics set using a configuration file (refer to Appendix D in the *Series IV Installation and Checkout Manual*). The configuration file is incorporated into the operating system using either the SYSGEN command (where the file is incorporated upon boot up) or the STTY command. Setting the system to multi-user mode can either be done in SYSGEN or with the REGION command. When serial channel 2 is used the Multi-user CH2 mode (in SYSGEN) should be selected.

Once the system is set up in the multi-user mode, the Series IV console and the terminal processes are independent from each other. However, the Background command cannot be used and only one user (configured in SYSGEN or REGION commands) may execute ISIS-IV. To exit the Multi-user mode, the second user must use the LOGOff exit option, then the REGION or SYSGEN commands must be used to reconfigure system mode.

2.12.3 Toggle Mode

The third mode of operation is the toggle mode, which may be established using either the SYSGEN or REGION commands. This mode is characterized by the ability of the Series IV to run two "foreground" processes at once. The key to the right of the F7 function key is the toggle key. This key determines which partition is interactive (displays on the console and accepts keyboard input). A message in the OS Field tells the user which partition is interactive (P1 or P2).

There are two options in toggle mode operation (Selected in SYSGEN or REGION). The first option allows the user to specify whether or not the screen is refreshed on every toggle. Although the option requires 4k of memory to be used, it is quite useful if performing screen oriented tasks in both regions. The second option allows the user to stop (control S) the console display when exiting a region and activate the console (control Q) when re-entering the region. An example of where this option would be useful would be in scrolling through a text file, while toggling to another process. Whenever the text file scroll is exited the screen freezes, preventing the user from missing any text that scrolls while the process is not being displayed.

REQUEST FOR READER'S COMMENTS

Intel's Technical Publications Departments attempt to provide publications that meet the needs of all Intel product users. This form lets you participate directly in the publication process. Your comments will help us correct and improve our publications. Please take a few minutes to respond.

Please restrict your comments to the usability, accuracy, readibility, organization, and completeness of this publication. If you have any comments on the product that this publication describes, please contact your Intel representative. If you wish to order publications, contact the Intel Literature Department (see page ii of this manual).

1. Please describe any errors you found in this publication (include page number).

2. Does the publication cover the information you expected or required? Please make suggestions for improvement.

3. Is this the right type of publication for your needs? Is it at the right level? What other types of publications are needed?

4. Did you have any difficulty understanding descriptions or wording? Where?

5. Please rate this publication on a scale of 1 to 5 (5 being the best rating.) ______

NAME ______ DATE ______

TITLE ______

COMPANY NAME/DEPARTMENT ______

ADDRESS ______

CITY ______ STATE (COUNTRY) ______ ZIP CODE ______

Please check here if you require a written reply. ☐

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) **987-8080**

Printed in U.S.A.

PE/0186/700 BJM

Development System